Conclusions and recommendations - i.e. how the site owner could improve the site security – with justifications. Recommendations should be ordered by business priority. **(Knowledge and Understanding weighted at 10%, Criticality weighted at 10%, Use of relevant sources weighted at 5%)**

**Site security recommendation notes for an e-commerce website -Need to be written up**:

Ying: To categories the below into HIGH, MEDIUM and LOW recommendations.

We can use the information written above for justification and examples.

Then each will write a paragraph on each category.

Ying: HIGH recommendations

Jon: MEDIUM recommendations

Haseeb: LOW recommendations

# Conclusions and recommendations

## Conclusions

Ying: Better to move here as a conclusion?

Consider ‘comparisons with the initial report, explanation of differences and omissions’ span/phishing etc if needed.

In the first phase, we provided some assumptions for potential vulnerabilities. Some of them are valid while others are not found, such as unsecured website / unsecured transactions because we don't have access and apply a black-box test. Therefore, we could not check if the data is encrypted before storing it as the website uses HTTPS and TLS 1.3 (appendix) certificates for communication. However, Burp Suite and Nessus found that passwords and logins are submitted in cleartext; DNS record does not contain Anti Spam/ Phishing policy; DOS/ DDOS attacks was excluded due to the 'house rules' not able to discovered through automated web scanners. We consider that the website was hosted by a good company that used WAF and used four subdomains in two different locations; this guaranteed protection and availability of service. Furthermore, the tools did not detect unrestricted access and weak authentication vulnerabilities. We could not get the login to the website by using the default user with common password list to bypass the login and crack the website credentials.

Ying: Suggest to remove

A key priority in our findings was injections attacks. According to OWASP (2021), injection attacks rank third in the top 10 list. Since 2017 such threats can include Cross Site Scripting (XXS) and SQL injection.

## Recommendations

### Short term

1. Use up-to-date software: Prevent using EOL software (Sugar Club, 2018). Consider to migrate an up-to-date CRM software and patching web and application server regularly to mitigate the impact of vulnerabilities.
2. Data encryption: Ensure that personal data and transactions are encrypted before transferring between the customer device and e-commerce website, it will help against a Man-In-The-Middle attack (Lokhande & Meshram, 2013). Force to use HTTPS with TLS encryption and restrict HTTP method, prevent cleartext submission for sensitive or critical information.
3. Account security: Limit unsuccessful login attempt by locking account and block IP address. Securing the server and admin panels by complex passwords and changing them frequently, using multi-factor authentication, and restricting access (inVerita, 2022).
4. Validate inputs: Implement a special characters filter to verify that only properly constructed data enters the process, which could help to reduce the impact of code injection significantly. (OWASP, N.D.)
5. SQL best practices: To prevent SQL injection attacks, Surkay (2021) suggested stopping dynamic SQL, storing sensitive credentials encrypted, limit database permissions and privileges on web application, and avoid showing database errors directly to the user.

### Medium term

1. Using cloud service which provided WAF, CDN or SIEM solutions to prevent malicious code injection and DoS attack.
2. Use Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHAs)
3. Vamsi (2021) mentioned that special characters need to be sanitised to prevent XSS attacks and disable external processing for entities.
   1. Use HttpOnly attribute cookies
   2. Anti-CSRF Tokens
4. Apply Domain-Based Message Authentication, Reporting & Conformance (DMARC) policy for Anti Spam/ Phishing.

### Long term

1. Staff and clients should be aware of the security policy and the potential threats; staff should not disclose their login information and password or share their credentials between them (Bader, 2021).
2. Live system vulnerability Scanning
   1. Lokhande & Meshram (2014) highlighted that developers and security professionals could fix the website by using an automated web vulnerability scanner to check the website to find out XSS vulnerabilities.
3. System design
   1. OWASP suggested remediation by separating data from commands using safe API and positive server-side validation. In addition, an intrusion detection system (IDS) could be incorporated for suspicious traffic on the client-side.

JON: Reduce the information here to key points.

The business impact has an enormous scope that could prove costly for companies using SugarCRM and SugarCRM itself. Any downtime, whether planned for maintenance or unplanned due to failure, will cause high costs. Therefore SugarCRM should implement the ITIL Service Value System (Nyhuis, 2020). This will set a framework for the company to provide the best service management practices and deliver business value. The framework underpins the ISO 20000 standard that demonstrates reliable and high-quality service, assures client confidence through compliance, and reduces costs of conformance to laws and standards. ISO 20000 encourages more detail in planning and monitoring for system configuration, incident management, change management and the relationships to any third party organisation or suppliers (IT Governance, N.D.). SugarCRM should aim for these best practices to build client relationships reputation and provide excellent service in the organisation's daily operation. Staff and clients would be aware of security policies to understand adequate security measures better. Effectively once the ISO 20000 standard is in place, SugarCRM would be in an excellent position to implement processes of Information Security Management and IT service continuity and availability. This framework would further enhance the security best practices by continually improving the Information Security Management System (ISMS) to implement policy, operation, monitoring, and review. This could be supported by the introduction of the ISO 27001 standard. ISO 27001 differs from service-based to risk-based management.

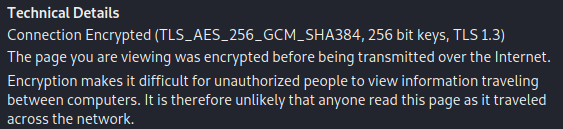
Whilst these recommendations can improve the security and service provided we must balance the cost. To meet the complete requirements of ISO 20000 and 27001 across SugarCRM would incur a high cost to implement. There would also be unplanned costs after the standard purchase, such as training and awareness, and staff resource time would affect the profitability and productivity. However, we must consider that the initial cost of investment in safeguarding benefits the value. For SugarCRM, this could support mitigation of the vulnerabilities that occurred. The risk approach of ISO 27001 would be helpful to SugarCRM should they want to not waste investments in information security (Boehmer, 2009).

SugarCRM could consider outsourcing their security needs. It could be a cost-effective method without the challenges of assembling the human resources, paying for training providers, purchasing necessary equipment and the expense of internal security measures. However, this does raise the level of risk by outsourcing, whereby the flow of the service product could become disrupted. Furthermore, outsourcing information security might lead to less expertise, meaning less control, less innovation, and organisational trust. When evaluating the cost, SugarCRM should consider accounting and decision-relevant costs.

Further concerns are raised concerning data protection. Any hired vendor handling sensitive data of SugarCRM and clients and protecting data and customer privacy should have processes and procedures in place. Management of access rights to the vendor would need to be carefully considered by SugarCRM.

# Appendix

Ying: To check SSL and screenshot of cleartext as evidence.

]

# References

OWASP. (N.D.) Input Validation Cheat Sheet. *OWASP Cheat Sheet Series*. Available from: https://cheatsheetseries.owasp.org/cheatsheets/Input\_Validation\_Cheat\_Sheet.html [Accessed 13 Feburary 2021].

Surkay, B. (January 27, 2021) What is SQL Injection and How to Prevent It?. *Vulnerabilities*. Available from: https://www.pcidssguide.com/what-is-sql-injection-how-to-prevent-it/ [Accessed 13 Feburary 2021].

inVerita. (January 31, 2022) Top 10 E-commerce Security Threats and their Solutions. *Blog*. Available from: https://inveritasoft.com/blog/top-ecommerce-security-threats-and-their-solutions [Accessed 13 Feburary 2021].

Lokhande, P. & Meshram, B. (2013) E-Commerce Applications: Vulnerabilities, Attacks and Countermeasures. *International Journal of Advanced Research in Computer Engineering & Technology*. Available from: https://www.researchgate.net/publication/235697382\_E-Commerce\_Applications\_Vulnerabilities\_Attacks\_and\_Countermeasures/link/0046351e3de4208a02000000/download [Accessed 13 Feburary 2021].